Appendix

A. Conditional Label Diffusion Models and Feature Introduction Analyzing

In this section, our main goal is to derive the posterior distribution of the forward diffusion process when image features are used as controlling conditions and to analyze its differences and connections with the unconditional posterior distribution. Following the method used in DDPM [6], We first define a conditional Markov process $\hat{q}$ (where q represents the unconditional label diffusion process in Section 3) in which Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is incrementally added to the labels for diffusion. The addition of noise remains the same whether features are conditioned or not, leading to the following definition:

$$\hat{q}(y_0) := q(y_0), \tag{1}$$

$$\hat{q}(y_t \mid y_{t-1}, x) := q(y_t \mid y_{t-1}), \tag{2}$$

$$\hat{q}(y_{1:T} \mid y_0, x) := \prod_{t=1}^{T} \hat{q}(y_t \mid y_{t-1}, x). \tag{3}$$

With knowledge of the forward process's prior distribution, we can derive the prior distribution of $\hat{q}$:

$$\begin{aligned}
\hat{q}(y_t \mid y_{t-1}) &= \int_x \hat{q}(y_t, x \mid y_{t-1})\, dx && (4)\\
&= \int_x \hat{q}(y_t \mid y_{t-1}, x)\, \hat{q}(x \mid y_{t-1})\, dx && (5)\\
&= \int_x q(y_t \mid y_{t-1})\, \hat{q}(x \mid y_{t-1})\, dx && (6)\\
&= q(y_t \mid y_{t-1}) \int_x \hat{q}(x \mid y_{t-1})\, dx && (7)\\
&= q(y_t \mid y_{t-1}) && (8)\\
&= \hat{q}(y_t \mid y_{t-1}, x), && (9)
\end{aligned}$$

which indicates that conditions do not affect the prior distribution in the forward process. Similarly, we can derive the joint distribution of $\hat{q}$:

$$\begin{aligned}
\hat{q}(y_{1:T} \mid y_0) &= \int_x \hat{q}(y_{1:T}, x \mid y_0)\, dx && (10)\\
&= \int_x \hat{q}(x \mid y_0)\, \hat{q}(y_{1:T} \mid y_0, x)\, dx && (11)\\
&= \int_x \hat{q}(x \mid y_0) \prod_{t=1}^{T} \hat{q}(y_t \mid y_{t-1}, x) dx && (12)\\
&= \int_x \hat{q}(x \mid y_0) \prod_{t=1}^{T} q(y_t \mid y_{t-1}) dx && (13)\\
&= \prod_{t=1}^{T} q(y_t \mid y_{t-1}) \int_x \hat{q}(x \mid y_0)\, dx && (14)\\
&= \prod_{t=1}^{T} q(y_t \mid y_{t-1}) && (15)\\
&= q(y_{1:T} \mid y_0). && (16)
\end{aligned}$$

Based on this result, we can further derive the marginal distribution of $\hat{q}$:

$$\begin{aligned}
\hat{q}(y_t) &= \int_{y_{0:t-1}} \hat{q}(y_0, \ldots, y_t)\, dy_{0:t-1} && (17)\\
&= \int_{y_{0:t-1}} \hat{q}(y_0)\, \hat{q}(y_1, \ldots, y_t \mid y_0)\, dy_{0:t-1} && (18)\\
&= \int_{y_{0:t-1}} q(y_0)\, q(y_1, \ldots, y_t \mid y_0)\, dy_{0:t-1} && (19)\\
&= \int_{y_{0:t-1}} q(y_0, \ldots, y_t)\, dy_{0:t-1} && (20)\\
&= q(y_t). && (21)
\end{aligned}$$

Using the prior and marginal distributions, we can demonstrate that the unconditional posterior distribution aligns with q:

$$\begin{aligned}
\hat{q}(y_{t-1} \mid y_t) &= \frac{\hat{q}(y_{t-1}, y_t)}{\hat{q}(y_t)} && (22)\\
&= \frac{\hat{q}(y_t \mid y_{t-1})\, \hat{q}(y_{t-1})}{\hat{q}(y_t)} && (23)\\
&= \frac{q(y_t \mid y_{t-1})\, q(y_{t-1})}{q(y_t)} && (24)\\
&= \frac{q(y_{t-1}, y_t)}{q(y_t)} && (25)\\
&= q(y_{t-1} \mid y_t). && (26)
\end{aligned}$$

By incorporating features as posterior conditions, we estimate the posterior distribution of the conditional forward process using Bayes' rule:

$$\hat{q}(y_{t-1} \mid x) = \frac{\hat{q}(y_{t-1})\, \hat{q}(x \mid y_{t-1})}{\hat{q}(x)}. \tag{27}$$

Continuing, by adding the known distribution y_t as a condition for generation, we can obtain:

$$\begin{aligned}
\hat{q}(y_{t-1} \mid y_t, x) &= \frac{\hat{q}(y_{t-1} \mid y_t)\, \hat{q}(x \mid y_{t-1}, y_t)}{\hat{q}(x|y_t)} && (28)\\
&= \frac{q(y_{t-1} \mid y_t)\, \hat{q}(x \mid y_{t-1}, y_t)}{\hat{q}(x|y_t)} && (29)\\
&= \frac{q(y_{t-1} \mid y_t)\, \hat{q}(x \mid y_{t-1})}{\hat{q}(x|y_t)} && (30)\\
&= q(y_{t-1} \mid y_t)\, e^{\log \hat{q}(x|y_{t-1}) - \log \hat{q}(x|y_t)}, && (31)
\end{aligned}$$

where the derivation of $\hat{q}\left(x \mid y_{t-1}, y_t\right) = \hat{q}\left(x \mid y_{t-1}\right)$ from Eq. 29 to Eq. 30 is as follows:

$$\hat{q}\left(x \mid y_{t-1}, y_t\right) = \hat{q}\left(y_t \mid y_{t-1}, x\right) \frac{\hat{q}\left(x \mid y_{t-1}\right)}{\hat{q}\left(y_t \mid y_{t-1}\right)} \tag{32}$$

$$= \hat{q}\left(y_t \mid y_{t-1}\right) \frac{\hat{q}\left(x \mid y_{t-1}\right)}{\hat{q}\left(y_t \mid y_{t-1}\right)} \tag{33}$$

$$= \hat{q}\left(x \mid y_{t-1}\right). \tag{34}$$

We note that the term $e^{-\log \hat{q}(x|y_t)}$ in Eq. 31 is independent of the distribution of y_{t-1}, thus we set this part as a constant C:

$$\hat{q}\left(y_{t-1} \mid y_t, x\right) = C \cdot q\left(y_{t-1} \mid y_t\right) e^{\log \hat{q}(x|y_{t-1})}, \tag{35}$$

where $q\left(y_{t-1} \mid y_t\right)$ is the unconditional posterior distribution of the diffusion process, modeled as a Gaussian distribution with mean $\tilde{\mu}_t\left(y_t, y_0\right)$ and variance $\tilde{\beta}_t$, respectively. Simplifying the covariance from the probability density formula, we can get:

$$\hat{q}\left(y_{t-1} \mid y_t, x\right) \propto e^{-\|y_{t-1}-\tilde{\mu}_t\|^2/2\tilde{\beta}_t + \log \hat{q}(x|y_{t-1})}. \tag{36}$$

Given that the number of time steps T in the diffusion process is large enough and the diffusion coefficient β_t is small enough, the variance of the distribution $\hat{q}\left(y_{t-1} \mid y_t\right)$ is sufficiently small and concentrated near $\tilde{\mu}_t$. We perform a Taylor expansion around $y_{t-1} = \tilde{\mu}_t$ for $\log \hat{q}\left(x \mid y_{t-1}\right)$ up to the first derivative, for simplicity, we let $\nabla_{y_{t-1}} \log \hat{q}\left(x \mid y_{t-1}\right)|_{y_{t-1}=\tilde{\mu}_t} = g$, which is essentially the gradient of the distribution at that point:

$$\log \hat{q}(x \mid y_{t-1}) \propto \log \hat{q}(x \mid y_{t-1})|_{y_{t-1}=\tilde{\mu}_t} \tag{37}$$

$$+ (y_{t-1} - \tilde{\mu}_t) g + \mathbf{o}(y_{t-1}). \tag{38}$$

Thus, the posterior distribution can be estimated as:

$$\hat{q}\left(y_{t-1} \mid y_t, x\right) \propto e^{-\|y_{t-1}-\tilde{\mu}_t\|^2/2\tilde{\beta}_t + (y_{t-1}-\tilde{\mu}_t)g + C_1} \tag{39}$$

$$\propto e^{-(\|y_{t-1}-\tilde{\mu}_t-\tilde{\beta}_t g\|^2)/2\tilde{\beta}_t + C_2} \tag{40}$$

$$= \mathcal{N}\left(y_{t-1}; \hat{\mu}_t, \sigma_t^2 \mathbf{I}\right), \tag{41}$$

where $\hat{\mu}_t = \frac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t} y_0 + \frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t} y_t + \sigma_t^2 \nabla_{y_{t-1}} \log \hat{q}(x \mid y_{t-1})$ and $\sigma_t = \sqrt{\tilde{\beta}_t} = \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}$. To ensure the correct introduction of conditions, we need to incorporate the decoding gradient into the mean during model prediction. Based on the minor nature of random perturbations (y_{t-1} vs. y_t) in the diffusion process [14], we define a decoding function $F_\phi(y_t, x, t) = \log p_\phi(x|y_t)$ to guide the mean shift, ensuring the dependency of label generation on image features.

B. Unconditional directional diffusion model

Based on the additivity of Gaussian noise, we derive the jump diffusion formula from y_0 to y_t without requiring single-step continuous diffusion:

$$y_t = y_{t-1} + \alpha_t y_d + \beta_t \epsilon_{t-1}, \tag{42}$$

$$= y_{t-2} + (\alpha_{t-1} + \alpha_t) y_d + (\sqrt{\beta_{t-1}^2 + \beta_t^2})\epsilon_{t-2} \tag{43}$$

$$= ... \tag{44}$$

$$= y_0 + \bar{\alpha}_t y_d + \bar{\beta}_t \epsilon, \tag{45}$$

where $\epsilon_{t-1}, \ldots \epsilon \sim \mathcal{N}(0, \mathbf{I})$, $\bar{\alpha}_t = \sum_{i=1}^{t} \alpha_i$ and $\bar{\beta}_t = \sqrt{\sum_{i=1}^{t} \beta_i^2}$. $\mathbf{y}_d$ is the deviation between $\mathbf{y}_n$ and y_0 (i.e., $\mathbf{y}_d = \mathbf{y}_n - \mathbf{y}_0$), representing the directional shift contained in the diffusion from $\mathbf{y}_{t-1}$ to $\mathbf{y}_t$. We define the forward distribution of the unconditional directed diffusion model as follows:

$$q\left(y_t \mid y_{t-1}, y_d\right) = \mathcal{N}\left(y_t; y_{t-1} + \alpha_t y_d, \beta_t^2 \mathbf{I}\right), \tag{46}$$

$$q\left(y_{1:T} \mid y_0, y_d\right) := \prod_{t=1}^{T} q\left(y_t \mid y_{t-1}, y_d\right) \tag{47}$$

$$= \mathcal{N}\left(y_t; y_0 + \bar{\alpha}_t y_d, \bar{\beta}_t^2 \mathbf{I}\right), \tag{48}$$

Similar to DDPM, we can represent the transfer probabilities $q(y_{t-1}|y_t, y_0, y_d)$ by Bayes' rule:

$$q(y_{t-1}|y_t, y_0, y_d) = q(y_t|y_{t-1}, y_0, y_d) \frac{q(y_{t-1}|y_0, y_d)}{q(y_t|y_0, y_d)}, \tag{49}$$

where $q(y_{t-1}|y_0, y_d) = \mathcal{N}(y_{t-1}; y_0 + \bar{\alpha}_{t-1} y_d, \bar{\beta}_{t-1}^2 \mathbf{I})$ and by the Markovian property of the forward process, y_t will not be affected by y_0, so we have $q(y_t|y_{t-1}, y_0, y_d) = q(y_t|y_{t-1}, y_d) = \mathcal{N}(y_t; y_{t-1} + \alpha_t y_d, \beta_t^2 \mathbf{I})$. Therefore, the posterior probability distribution of the forward process can be derived as follows:

$$q(y_{t-1}|y_t, y_0, y_d) = \mathcal{N}(y_{t-1}; \mu_t, \sigma_t \mathbf{I}), \tag{50}$$

$$\propto \exp\left(-\frac{1}{2}(A y_{t-1}^2 - 2B y_{t-1} + C)\right), \tag{51}$$

where $A = \frac{\bar{\beta}_t^2}{\beta_t^2 \bar{\beta}_{t-1}^2}$, $B = \frac{y_t - \alpha_t y_d}{\beta_t^2} + \frac{\bar{\alpha}_{t-1} y_d + y_0}{\bar{\beta}_{t-1}^2}$ and $C(y_t, y_0, y_d)$ is not related to y_{t-1}. Then, we can get μ_t and σ_t in Eq. 50:

$$\mu_t = B/A \tag{52}$$

$$= \frac{\bar{\beta}_{t-1}^2}{\bar{\beta}_t^2} y_t + \frac{\beta_t^2 \bar{\alpha}_{t-1} - \bar{\beta}_{t-1}^2 \alpha_t}{\bar{\beta}_t^2} y_d + \frac{\beta_t^2}{\bar{\beta}_t^2} y_0 \tag{53}$$

$$= y_t - \alpha_t y_d - \frac{\beta_t^2}{\bar{\beta}_t} \epsilon, \tag{54}$$

$$\sigma_t = 1/A = \frac{\beta_t^2 \bar{\beta}_{t-1}^2}{\bar{\beta}_t^2}. \tag{55}$$

In the reverse process, we define the predictive distribution of the model as $p_\theta(y_{t-1}|y_t)$. The diffusion model is learned by optimizing the evidence lower bound with stochastic gradient descent:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q \left[\mathcal{L}_T + \sum_{t>1}^{T} \mathcal{L}_{t-1} + \mathcal{L}_0 \right]. \tag{56}$$

Following [6], we only consider L_{t-1}:

$$L_{t-1} = D_{KL}(q(y_{t-1}|y_t, y_0, y_d)||p_\theta(y_{t-1}|y_t)) \tag{57}$$

$$= \mathbb{E}\left[\left\| y_t - \alpha_t y_d - \frac{\beta_t^2}{\bar{\beta}_t}\epsilon - (y_t - \alpha_t y_d^{\theta_1} - \frac{\beta_t^2}{\bar{\beta}_t}\epsilon_{\theta_2}) \right\|^2 \right], \tag{58}$$

where D_{KL} denotes KL divergence. $y_d^{\theta_1}$ and ϵ_{θ_2} represent the prediction of orientation difference and noise with two networks, respectively. By reparameterizing, the objective can eventually be simplified to:

$$\mathcal{L}_d = || \mathbf{y}_d - \mathbf{y}_d^{\theta_1}(\mathbf{y}_t, \mathbf{y}_n, \mathbf{x}, t) ||^2, \tag{59}$$

$$\mathcal{L}_\epsilon = || \epsilon - \epsilon_{\theta_2}(\mathbf{y}_t, \mathbf{y}_n, \mathbf{x}, t) ||^2. \tag{60}$$

C. Conditional directional diffusion model

This Section mainly combines the relevant theories of conditional label diffusion model (Section A) and unconditional directional diffusion model (Section B), and analyzes the feature introduction method of conditional directed label diffusion model, which is also the theory that our DLD's network architecture mainly relies on. Similar to conditional label diffusion model, We define the conditional forward distribution $\hat{q}$ as follows:

$$\hat{q}(y_0) := q(y_0), \tag{61}$$

$$\hat{q}(y_t \mid y_{t-1}, y_d, x) := q(y_t \mid y_{t-1}, y_d), \tag{62}$$

$$\hat{q}(y_{1:T} \mid y_0, x) := \prod_{t=1}^{T} \hat{q}(y_t \mid y_{t-1}, y_d, x) \tag{63}$$

$$= \prod_{t=1}^{T} q(y_t \mid y_{t-1}, y_d, x) \tag{64}$$

$$= \prod_{t=1}^{T} q(y_t \mid y_{t-1}, y_d). \tag{65}$$

This means that the diffusion process of the label is not affected by any image features, and the related theory of its forward process is consistent with Eq. 42 to Eq. 48. To estimate the posterior distribution $q(y_{t-1} \mid y_t, y_d, x)$, similarly to the analysis of the conditional label diffusion model, we separate it into the relevant and irrelevant components of the image feature x using the Bayes' rule:

$$\hat{q}(y_{t-1} \mid y_t, y_d, x) = \hat{q}(y_{t-1} \mid y_t, y_d) \frac{\hat{q}(x \mid y_{t-1}, y_t, y_d)}{\hat{q}(x|y_t, y_d)} \tag{66}$$

$$= q(y_{t-1} \mid y_t, y_d) \frac{\hat{q}(x \mid y_{t-1}, , y_d)}{\hat{q}(x|y_t, y_d)} \tag{67}$$

$$= C \cdot q(y_{t-1} \mid y_t, y_d)\, \hat{q}(x \mid y_{t-1}, y_d), \tag{68}$$

where C represents $\hat{q}(x|y_t, y_d)^{-1}$, which means a constant term independent of the distribution of y_{t-1}. Then we use the probability density function of the Gaussian distribution and the Taylor expansion to obtain the exponential form:

$$q(y_{t-1} \mid y_t, y_d, x) \propto e^{-\|y_{t-1} - \mu_t\|^2 / 2\sigma_t + \log \hat{q}(x|y_{t-1}, y_d)} \tag{69}$$

$$= \mathcal{N}(y_{t-1}; \hat{\mu}_t, \hat{\sigma}_t \mathbf{I}), \tag{70}$$

$$\hat{\mu}_t = \mu_t + \hat{\sigma}_t \nabla_{y_{t-1}} \log \hat{q}(\mathbf{x}|\mathbf{y}_{t-1}, \mathbf{y}_d) \tag{71}$$

$$= \frac{\bar{\beta}_{t-1}^2}{\bar{\beta}_t^2}\mathbf{y}_t + \frac{\beta_t^2 \bar{\alpha}_{t-1} - \bar{\beta}_{t-1}^2 \alpha_t}{\bar{\beta}_t^2}\mathbf{y}_d + \frac{\beta_t^2}{\bar{\beta}_t^2}\mathbf{y}_0 \tag{72}$$

$$+ \hat{\sigma}_t \nabla_{y_{t-1}} \log \hat{q}(\mathbf{x}|\mathbf{y}_{t-1}, \mathbf{y}_d) \tag{73}$$

$$= y_t - \alpha_t y_d - \frac{\beta_t^2}{\bar{\beta}_t}\epsilon + \hat{\sigma}_t \nabla_{y_{t-1}} \log \hat{q}(\mathbf{x}|\mathbf{y}_{t-1}, \mathbf{y}_d) \tag{74}$$

$$\hat{\sigma}_t = \sigma_t = \frac{\beta_t^2 \bar{\beta}_{t-1}^2}{\bar{\beta}_t^2}. \tag{75}$$

To ensure the correct introduction of conditions, we define a decoding function $F_\phi(y_t, y_d, x, t) = \log p_\phi(x|y_t, y_d)$ to guide the mean shift, ensuring the dependency of label generation on image features. The diffusion model's

network architecture, depicted in Figure C.1, consists of a ResNet encoder and a series of feedforward layers. The L_1 decoding layer plays a crucial role by contributing the $F_\phi(y_t, y_d, x, t)$ as guidance. The network inputs are (x, y_0, y_n), randomly sampled t and ϵ, where y_0 and y_n is transformed into y_t by Eq. 45 and then concatenated with $f_{ext}(x)$. After L_1's decoding, it merges with the normalized encoding features of ResNet through a Hadamard product, incorporates time positional encoding, and uses a series of feedforward networks, batch normalization, and Softplus activation to predict the directional deviation $y_d^{\theta_1}$ or noise term ϵ_{θ_2}.

D. Deterministic Implicit Inference

This section primarily analyzes the inference process of the DLD algorithm. Since the diffusion process involves labels and is directed towards classification tasks, it is imperative to reduce the uncertainty in the inference process and expedite it as much as possible, aligning with the method of denoising diffusion implicit models (DDIM) [16]. With a trained directional diffusion model, we proceed as follows:

$$q\left(y_t \mid y_0, y_d\right) = \mathcal{N}\left(y_t; y_0 + \bar{\alpha}_t y_d, \bar{\beta}_t^2 \mathbf{I}\right), \tag{76}$$

$$y_t = y_0 + \bar{\alpha}_t y_d + \bar{\beta}_t \epsilon, \tag{77}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. Similar to DDIM, we define a non-Markovian nature for the forward posterior distribution $q_{\sigma_k}\left(y_{\tau_{k-1}} \mid y_{\tau_k}, y_0, y_d\right) = \mathcal{N}\left(y_{\tau_{k-1}}; y_0 + M y_{\tau_k} + N y_d, {\sigma_k}^2 \mathbf{I}\right)$, where M and N are coefficients to be determined, and $\sigma_k \geq 0$. τ is a subsequence of $[1, \cdots, T]$, with $\tau_K = T$, e.g., if $T = 1000$ and $K = 10$, then $\tau = [1, 100, \cdots, 900, 1000]$. From the empirical form of the posterior distribution, we have:

$$\begin{aligned} y_{\tau_{k-1}} &= A y_0 + B y_d + C y_{\tau_k} + \sigma_k \epsilon && (78)\\ &= A y_0 + B y_{\tau_k} + C(y_0 + \bar{\alpha}_{\tau_k} y_d + \bar{\beta}_{\tau_k} \dot{\epsilon}) + \sigma_k \epsilon && (79)\\ &= (A + C) y_0 + (B + C\bar{\alpha}_{\tau_k}) y_d + \sqrt{C^2 \bar{\beta}_{\tau_k}^2 + \sigma_k^2}\epsilon && (80)\\ &= y_0 + \bar{\alpha}_{\tau_{k-1}} y_d + \bar{\beta}_{\tau_{k-1}} \dot{\epsilon}, && (81) \end{aligned}$$

By the method of undetermined coefficients, A, B and C are determined:

$$A = 1 - \frac{\sqrt{\bar{\beta}_{\tau_{k-1}}^2 - \sigma_k^2}}{\bar{\beta}_{\tau_k}}, \tag{82}$$

$$B = \bar{\alpha}_{\tau_{k-1}} - \frac{\sqrt{\bar{\alpha}_{\tau_k} \bar{\beta}_{\tau_{k-1}}^2 - \sigma_k^2}}{\bar{\beta}_{\tau_k}}, \tag{83}$$

Algorithm D.1 DLD inference

Input: Testing set $\mathcal{D} = \{\mathbf{X}\}$, Two trained network f_{θ_1} and f_{θ_2}

Output: y_0

1: Sample data $x \sim \mathcal{D}$, Sample label $y_T \sim \mathcal{N}(0, \mathbf{I})$
2: **for** $k = K$ **to** 1 **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $k > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad$ Predict $y_d^{\theta_1}$ and ϵ_{θ_2} using networks
5: $\quad$ Inference $y_{\tau_{k-1}} = y_{\tau_k} - (\bar{\alpha}_{\tau_k} - \bar{\alpha}_{\tau_{k-1}}) y_d^{\theta_1} - (\bar{\beta}_{\tau_k} - \sqrt{\bar{\beta}_{\tau_{k-1}}^2 - \sigma_k^2})\epsilon_{\theta_2} + \sigma_k \mathbf{z}$
6: **end for**

$$C = \frac{\sqrt{\bar{\beta}_{\tau_{k-1}}^2 - \sigma_k^2}}{\bar{\beta}_{\tau_k}}. \tag{84}$$

We reorganize the inference distribution, and since the y_0 term is unknown during actual inference, we estimate it using $y_0 = y_{\tau_k} - \bar{\alpha}_{\tau_k} y_d^{\theta_1} - \bar{\beta}_{\tau_k} \epsilon_{\theta_2}$. The inference process can be simplifying as:

$$\begin{aligned} y_{\tau_{k-1}} = y_{\tau_k} &- (\bar{\alpha}_{\tau_k} - \bar{\alpha}_{\tau_{k-1}}) y_d^{\theta_1} \\ &- (\bar{\beta}_{\tau_k} - \sqrt{\bar{\beta}_{\tau_{k-1}}^2 - \sigma_k^2})\epsilon_{\theta_2} + \sigma_k \mathbf{z} \end{aligned} \tag{85}$$

The detailed inference process is outlined in Algorithm D.1. For classification tasks, following the DDIM approach, we can achieve an implicit probabilistic diffusion model, turning the inference into a deterministic process given y_T by setting $\sigma_k = 0$ [13]. This modification reduces the variability during inference, ensuring more consistent and reliable label predictions crucial for classification accuracy. To better illustrate the inference process of DLD, we visualized its 5-step classification performance on the CIFAR-10 dataset using t-SNE, based on the feature space of ViT (see Figure D.1). Compared to LRA-diffusion, our inference is faster, more accurate, and results in fewer incorrect predictions and clearer category boundaries after inference completion.

E. Experimental Setup and Details

E.1. Generation of Instance-dependent Noise

We conducted experiments on the CIFAR-10 and CIFAR-100 datasets with simulated label noise. To closely mimic the distribution of label noise in real-world scenarios, we uesd a deep neural network to simulate instance-dependent noise (IDN) [2]. Specifically, we trained a WideResNet [19] for T epochs and recorded the average prediction for each sample throughout the training process as $S^t = [f^t(x_i)]_{i=1}^n \in \mathbb{R}^{n \times c}$. Then we calculated the mislabeling score for each sample as $N(x_i) = max_{k \neq y_i} S_{i,k}$ and the potential noise label as $\tilde{y}(x_i) = argmax_{k \neq y_i} S_{i,k}$, introducing label noise to the top $p\%$ of samples based on the

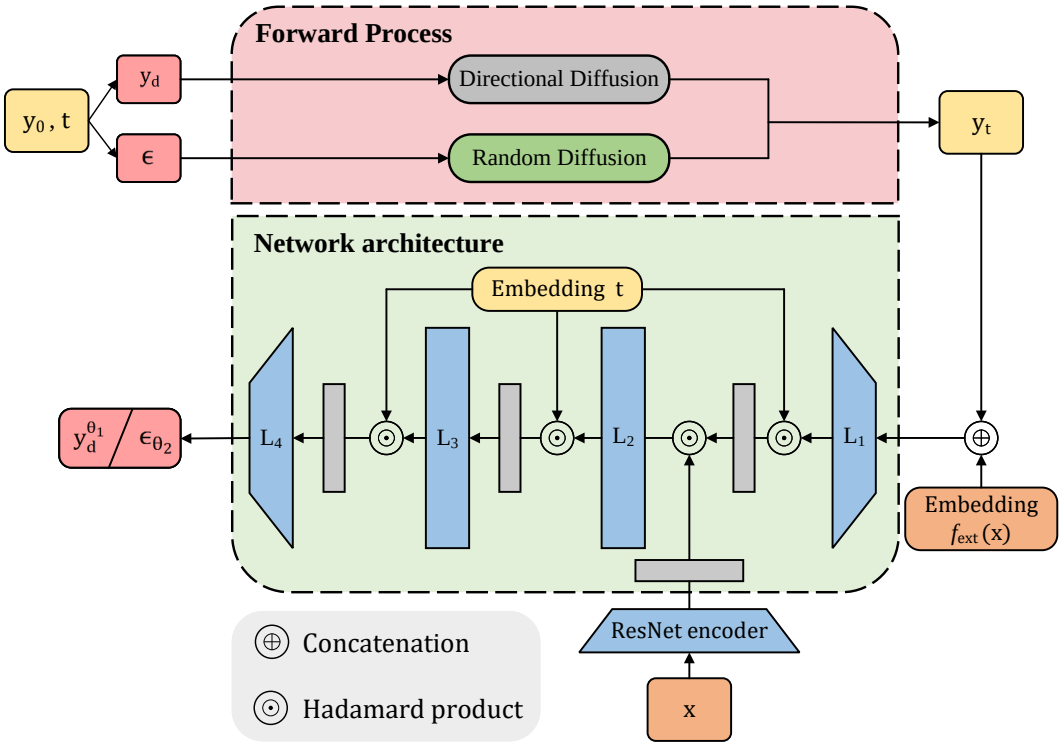


Figure C.1. The network architecture of the directional diffusion model. The input to the network consists of four elements: $\mathbf{y}_0$, t (represented by yellow blocks), $\mathbf{x}$ and $f_{ext}(\mathbf{x})$ (represented by orange blocks). Blue components represent trainable network layers, while gray components represent normalization activation layers. When a dual-network architecture is employed, the output of each network is predicted as random noise or directional deviation (represented by pink blocks). In contrast, when a single network is used to accelerate training, it outputs both random noise and directional bias simultaneously through two channels.

Algorithm D.2 IDN Generation

Input: Clean dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, a targeted noise fraction p, epochs T
Initialize a network f
for $t = 1$ to T **do**
  for batches $(x_i, y_i)_{i\in\mathcal{B}}$ **do**
    Train f on $(x_i, y_i)_{i\in\mathcal{B}}$ using cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \log\left(f_{y_i}^t(x_i)\right)$$

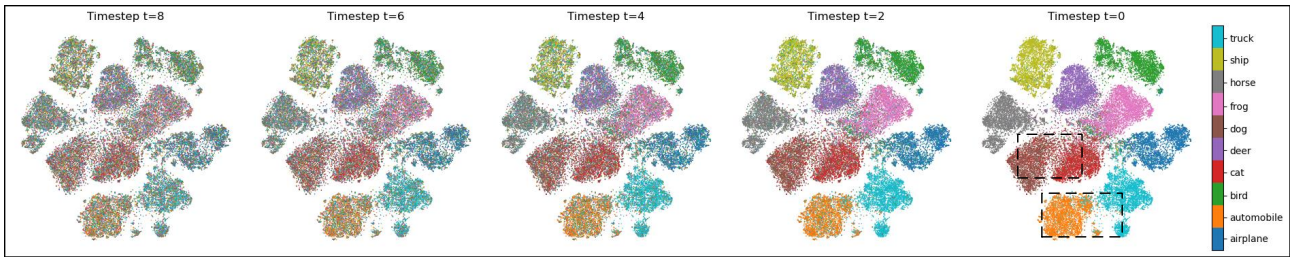
  end for
  Record output $S^t = [f^t(x_i)]_{i=1}^n \in \mathbb{R}^{n\times c}$
end for
Compute $N(x_i)$, $\tilde{y}(x_i)$ using $\{S^t\}_{t=1}^T$
Compute the index set $I = p\%\text{argmax}_{1\le i\le n} N(x_i)$
Flip $y_i = \tilde{y}_i$ if $i \in I$, else keep $y_i = y_i$
return a dataset with IDN: $\tilde{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^n$

mislabeling score. The algorithm D.2 details the process for simulating IDN, indicating that samples more prone to prediction errors are more likely to be labeled as noise, and the noise labels originate from the category most similar to their features.

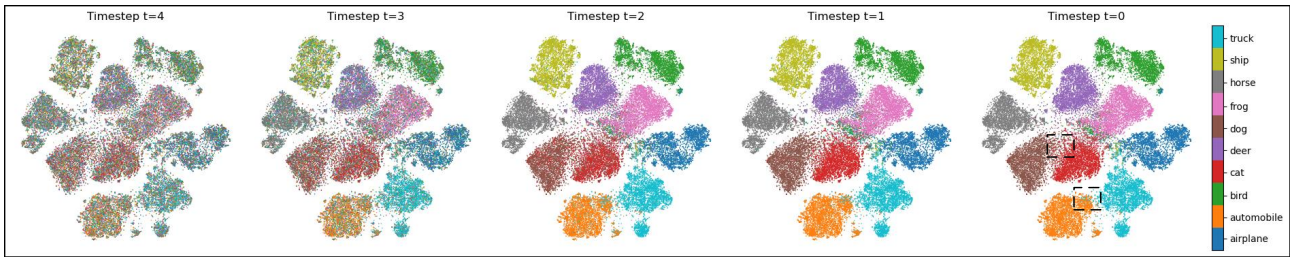
As depicted in Figure E.1, the experimentally simulated noise labels are highly instance-dependent in the CIFAR-10 dataset, with some pictures also being difficult for the human eye to distinguish. For instance, most of the *automobile* instances in the second row are incorrectly labeled as *trucks*, while a green *automobile* in the fourth column is labeled as a *frog*. Conversely, the *truck* samples in the last row are mostly labeled as *automobile*. This indicates that the IDN in our experiment has a degree of confusion approximating real-world noise, and such a simulated noise environment can better reflect the real performance of the robust learning model.

E.2. Experimental Results on CIFAR with Class-conditional Noise

Although class-conditional noise (CCN) is easier to handle than IDN noise, we also conducted experiments on simulated CCN datasets to further demonstrate the comprehensiveness of the DLD model. The noise settings in the experiment are identical to those in DivdeMix [10]. The ex-



(a) Inference process demonstration of LRA-diffusion using 10-step DDIM



(b) Inference process demonstration of DLD using 5-step DDIM

Figure D.1. Comparison of inference processes for different diffusion models on the CIFAR-10 dataset (with 40% IDN), visualized using t-SNE [17] on the ViT feature space. Data points are color-coded according to the highest probability value in the intermediate/final label vectors, corresponding to their respective categories. The black dashed box highlights regions with significant inference errors.

perimental results are presented in Table E.1. As shown in Figure E.2, we visualized three noise distributions (i.e., IDN, Symmetric, Asymmetric) on CIFAR-10 using t-SNE [17] with ViT feature spaces. It is evident that, at the same noise ratio (40%), the IDN contamination is more concentrated in categories with greater similarity, such as *truck* vs. *automobile* and *cat* vs. *dog* (see Figure E.2b), resulting in visibly higher contamination compared to Symmetric and Asymmetric noise. Notably, in simulating Asymmetric noise, the sample labels are flipped in only one direction (e.g., *truck* to *automobile*), meaning the actual noise ratio does not reach the intended value (see Figure E.2d). DLD consistently maintains superior performance against symmetric and asymmetric noise of various proportions. This is particularly evident when the number of categories is large and the noise proportion is high (e.g., CIFAR100 with 80% CCN). Its classification accuracy shows significant advantages compared to other algorithms.

E.3. Training Efficiency Analysis

The time overhead and model parameter comparison in Figure E.3 highlight the significant cost-efficiency of our DLD. Compared to collaborative or iterative training architectures like CC and DivideMix, DLD accelerates training by 40% to 100% and reduces model parameters by 30%, which is groundbreaking. Furthermore, we can reduce the dual-network model to a single-network model with dual-channel output, maintaining performance advantages while lowering the training cost to the LRA-diffusion baseline. Detailed settings are available in F.1. The high efficiency and resource savings of DLD arise from its sample pre-correction process, which can be pre-computed and cached before training, eliminating the need for redundant calculations. In other words, it is a one-time process, meaning that when handling large-scale datasets, its time cost is negligible compared to model training, yet its contribution to enhancing classification performance is significant. In addition, the model architecture of DLD is lightweight enough, which can be seen from its number of parameters.

E.4. Real-world Dataset Details

Animal-10N consists of 60,000 images, with 50,000 in the training set and 5,000 in the test set. These images are sourced from Google and Bing searches of five pairs of visually similar animals, such as cats and servals. Due to the resemblance between these category pairs, there is a high likelihood of confusion, resulting in approximately 8% mislabeling in the training set. However, the samples in the test set have been meticulously inspected by experts and carry accurate and reliable labels.

Clothing1M includes 1 million clothing images from various online shopping sites. These images are automatically categorized into 14 classes based on keywords in the surrounding text, with about 38.5% of the labels be-

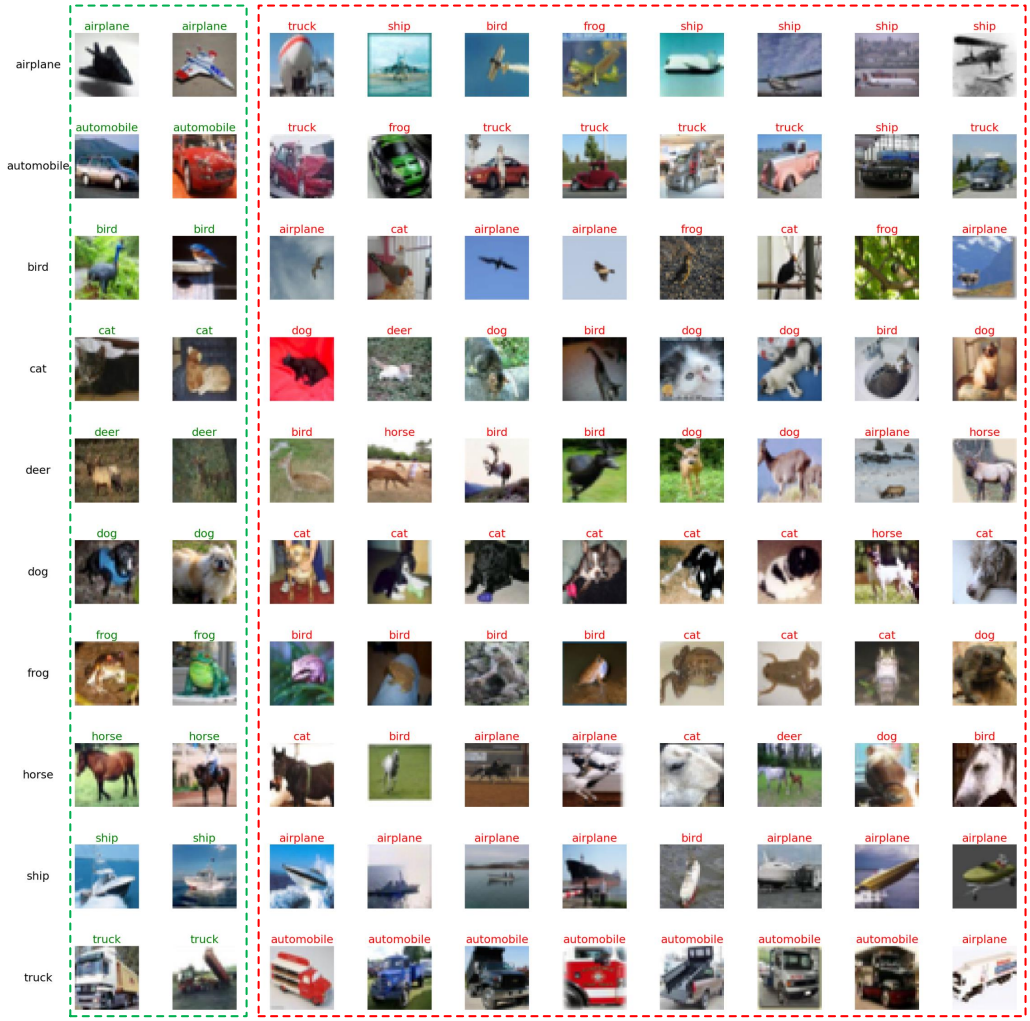
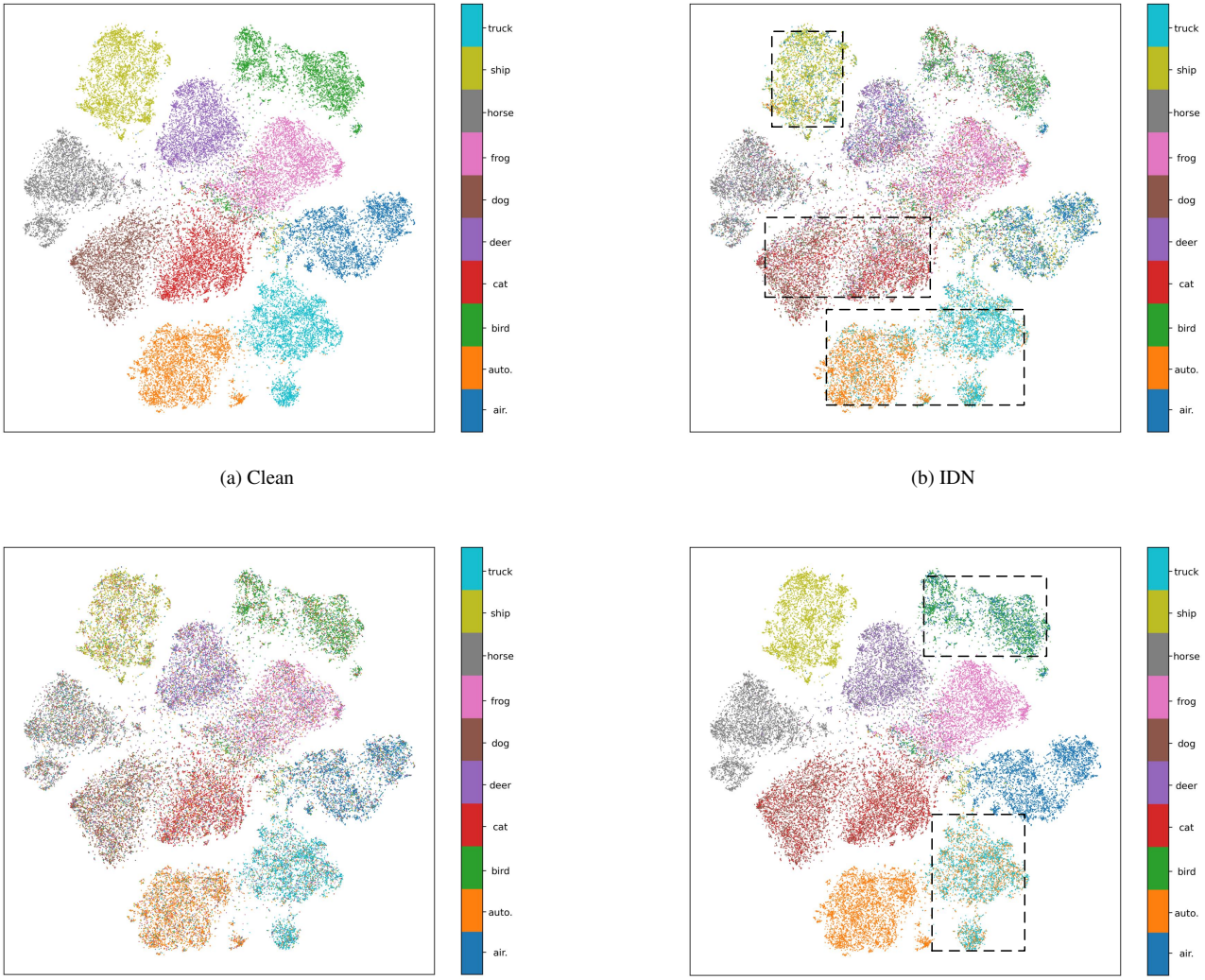


Figure E.1. IDN labels (on the top of each image) generated on CIFAR-10. Each row corresponds to instances of the same true class, with the first two columns representing correct examples without induced noise, and the last eight columns showing instances with noisy labels.

Table E.1. Comparison with state-of-the-art methods in test accuracy(%) on CIFAR with class-conditional noise.

Dataset	CIFAR10					CIFAR100			
Noise Type	Sym.				Asym.	Sym.			
Noise Rate	20%	40%	60%	80%	40%	20%	40%	60%	80%
CARD [5]	86.81	80.47	77.31	62.90	85.01	62.07	48.73	44.57	21.17
Co-teaching [4]	89.53	86.75	83.57	67.43	87.53	65.65	53.38	49.87	27.93
GCE [20]	92.40	91.11	87.24	77.51	88.51	69.41	59.54	55.32	31.19
DMI [18]	94.07	93.06	90.37	86.83	87.47	73.91	68.66	64.38	48.25
DivideMix [10]	96.14	95.64	93.67	93.24	93.46	77.32	76.67	71.06	60.24
ELR+ [12]	95.83	95.89	92.61	93.33	93.01	77.65	75.43	72.51	60.80
EPL [9]	96.13	96.07	95.99	95.89	95.01	77.65	77.13	76.22	72.07
LRA-diffusion [1]	96.31	96.27	95.82	95.76	95.37	78.01	77.59	76.34	73.57
DLD (Ours)	**97.22**	**97.15**	**97.11**	**96.48**	**97.13**	**78.96**	**78.21**	**78.08**	**76.82**



(a) Clean

(b) IDN

(c) Symmetric noise

(d) Asymmetric noise

Figure E.2. Comparison of t-SNE [17] visualization on CIFAR10 with different noise distributions at 40% noise rates. The black dashed box highlights regions with significant noise contamination.

ing noisy. The dataset also contains a clean subset of the training, validation, and test sets, which have been manually refined and contain approximately 47.6k, 14.3k, and 10k images, respectively. However, we opt not to use the clean training data and instead use only the noisy labeled data for model training.

WebVision and ILSVRC2012 feature a total of 2.4 million images collected from Google and Flickr searches based on the ILSVRC12 classification system. Building on previous research, we selected the top 50 classes from the Google image subset for model training and evaluated performance on the validation sets of both WebVision and ILSVRC12.

E.5. Implementation Details

In our experiments, we configured ResNet34 and ResNet50 (depicted as blue ResNet blocks in Figure C.1) as trainable encoders for the CIFAR datasets and real-world datasets, respectively. For the CIFAR dataset experiments, all forward layers were set to a dimension of 512, while for the real-world datasets, the dimension was set to 1024. We trained the networks for 200 epochs using the Adam optimizer with a batch size of 256. The initial learning rate was set at 0.001, with an adaptive learning rate adjustment strategy that included a warm-up phase and a half-cycle cosine decay phase. For the diffusion coefficients α_t and β_t, similarly to RDDM [11], we set α_t to linearly decrease over

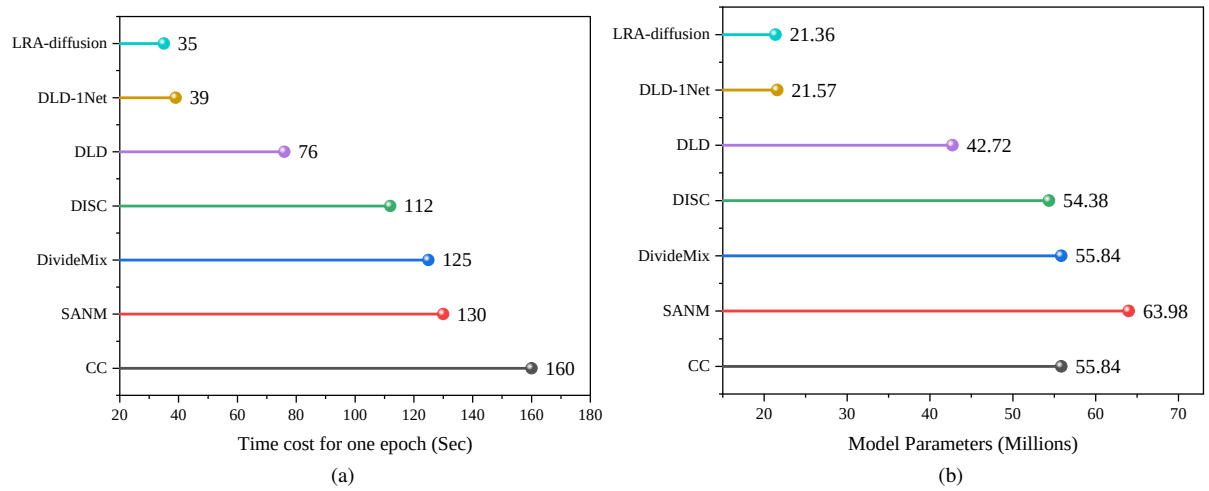


Figure E.3. Comparison of the training time (sec) and model parameter count (Millions) for different LNL methods. The results for each method are averaged over one epoch of training on the CIFAR-10 dataset using an NVIDIA A800 GPU.

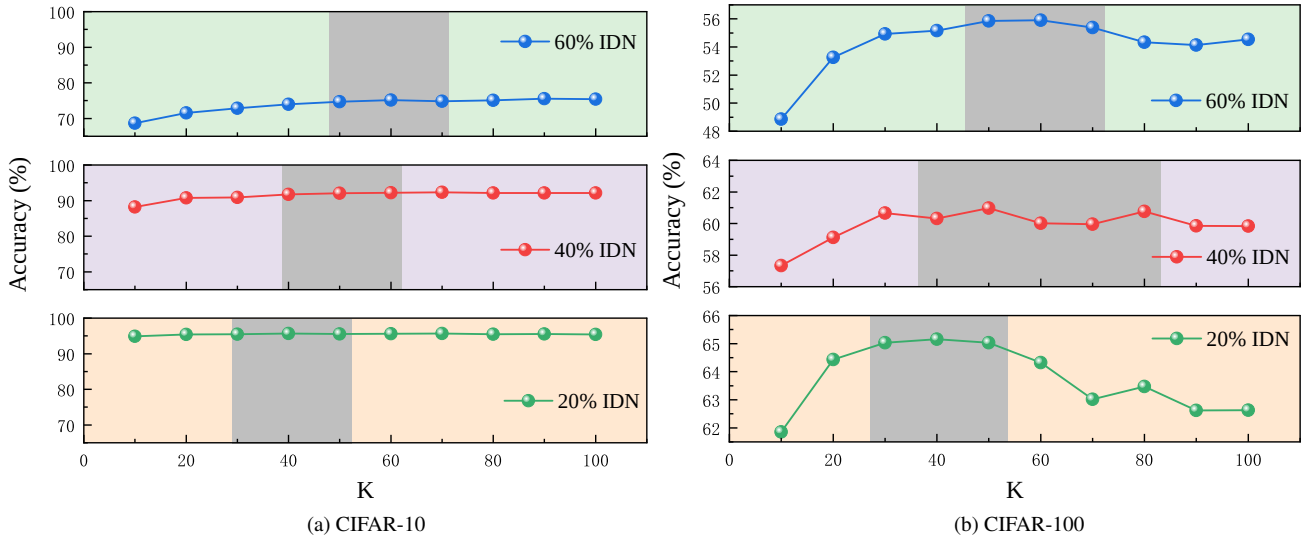


Figure E.4. The effect of different K values on the classification accuracy (%) of the DLD model under various IDN levels on the CIFAR dataset. The gray background highlights the range of K values where the accuracy is relatively stable and exhibits robust performance across different IDN levels.

time t and β_t to linearly increase, as described by the following formula:

$$P(x, a) := x^a / \int_0^1 x^a dx, \text{where} x = t/T. \quad (86)$$

Similar to FixMatch [15], we employed both weak and strong augmentations to create two different views of the dataset during the sample repartitioning phase. Weak augmentation included resizing, random horizontal flipping, and random cropping, while strong augmentation employed the random augmentation ($n = 2$, $m = 10$) technique [3], which randomly selects two out of fifteen augmentation techniques and applys them to the images with 10% intensity.

To analyze the impact of K values on model performance, we conducted tests on the CIFAR dataset under varying IDN levels with K ranging from 1 to 100. Our experiments showed that the DLD's accuracies remained relatively stable for k values between 30 and 70 (see Figure E.4). Additionally, we observed that as the noise ratio increases, the optimal value of K also rises. However, excessively large values of K increase computational costs without yielding significant improvements in model performance. Considering these factors, we set $K = 50$ as the de-

fault setting for DLD, as it achieves both stability and high accuracy across various noise conditions. This phenomenon can be explained by the need to expand the neighborhood size to stabilize label distributions when noisy information becomes prevalent, thereby improving the model's robustness. All experiments were conducted on eight NVIDIA A800 GPUs.

F. More Details of Ablation Studies

F.1. Cross ablation experiments

We designed a cross-ablation experiment to further analyze the contributions of the two stages in the DLD framework: the DLD model and the pre-correction method. The experimental results, as shown in Table F.1, indicate that alternative configurations exhibit a performance gap compared to the complete DLD framework (pre-correction method with DLD dual-network model). Among these, the DLD-1Net configuration shows the smallest gap, indicating that a single network can reduce computational overhead while maintaining performance. The primary difference between LRA + DLD w/o pre-correction and LRA-diffusion lies in the choice of the diffusion model (DLD dual-model vs. standard diffusion model). Results show that, under otherwise consistent conditions, the DLD dual-model improves noise robustness more effectively. The primary difference between pre-correction + LRA-diffusion w/o LRA and LRA-diffusion lies in the use of the pre-correction method. While pre-correction improves the performance of the standard diffusion model, its effect is less significant than that of replacing it with the DLD model, suggesting that the contribution of DLD outweighs the pre-correction strategy. In conclusion, the combination of the two stages leads to more significant improvements. The DLD model is able to leverage more accurate diffusion information to achieve a greater effect.

F.2. Contribution of Pre-trained Features

We initially conducted research on different settings of the pre-trained feature extractor f_{ext} and performed ablation experiments comparing our method with DISC, CARD, and LRA-diffusion. Both the CARD and LRA-diffusion models require the integration of a pre-trained model (CARD need a pre-trained classifier), using f_{ext} to guide the forward and reverse processes, whereas DISC is an optimal semi-supervised LNL method that does not require any pre-trained models. We selected three pre-trained feature extractors for our experiments:

- ResNet34: two ResNet34 architectures pre-trained through self-supervised contrastive learning on unlabeled datasets;
- SimCLR: a self-supervised model pre-trained on millions of unlabeled images using contrastive learning. It leverages a large ResNet backbone with millions of parameters to learn powerful visual representations;
- ViT-L/14: a vision transformer (ViT) model with 306 million parameters, pre-trained on the Imagenet dataset containing over 4 million image-text pairs, providing our framework with exceptional feature extraction capabilities.

Table F.2 shows that when all three models use the same pre-trained feature extractor, DLD consistently outperforms the other two methods, indicating that its superior performance arises from capabilities beyond feature extraction. When combined with ViT-L/14, all three methods perform optimally, suggesting that ViT-L/14's ability to map data into latent space provides more precise label information for the diffusion model to learn from. The classification performance of DLD combined with the three feature extractors exceeds that of DISC, although the advantage decreases as the extractors' quality diminishes. The performance of ResNet34 as a feature extractor is the weakest. When combined with it, the performance of the other two models, except DLD, is negatively affected by the poor-quality features, leading to classification accuracy close to or below that of DISC. Therefore, in subsequent experimental setups, we default to using ViT-L/14 as the pre-trained feature extractor to enhance the generalization ability of DLD.

F.3. Other Hyperparameter Analysis

To further validate the effectiveness of various settings in the DLD training strategy, we used the results of LRA-diffusion as the baseline on the CIFAR-100 dataset and compared different DLD training strategies. These strategies included neighbor selection strategies (using KNN, or using cosine similarity for neighbor label distribution collection), label pre-correction strategies (converting the pre-corrected label results into one-hot labels through the argmax function, or maintaining original vector labels), and different use of enhanced image features (using features of a single view only or using fusion features of two views as generation conditions). We conducted four sets of control experiments with the following detailed settings:

- DLD-knn: Using euclidean distance as the KNN label distribution collector, converting pre-corrected labels to one-hot labels, and using features of a single view only as generation conditions;
- DLD-cos: Using cosine similarity as the the KNN label distribution collector, with other settings identical to DLD-knn;
- DLD-vector: Retaining vector labels during label pre-correction, with other settings identical to DLD-cos;
- DLD-ws: Using the mean of image features from two views as generation conditions, with other settings identical to DLD-vector.

As shown in Figure F.1, in low noise environments

Table F.1. Results of the cross-ablation experiment on the CIFAR datasets with 40% IDN. DLD-1Net refers to using all components of the proposed DLD framework (pre-correction method and DLD model) but with a single network architecture, where one network outputs both the ϵ and $\mathbf{y}_d$ channels. LRA + DLD w/o pre-correction refers to replacing the pre-correction method in the DLD framework with the LRA strategy. Pre-correction + LRA-diffusion w/o LRA refers to replacing the pre-correction method in the DLD framework with the pre-correction method from LRA-diffusion.

Dataset	CIFAR-10		CIFAR-100	
Method	Accuracy	Gap	Accuracy	Gap
DLD	96.49	-	76.03	-
DLD-1Net	96.49	0.00 ↓	75.91	0.12 ↓
LRA + DLD w/o pre-correction	96.03	0.46 ↓	74.37	1.54 ↓
pre-correction + LRA-diffusion w/o LRA	95.11	0.92 ↓	71.63	2.74 ↓
LRA-diffusion	93.68	1.73 ↓	67.67	4.96 ↓

Table F.2. Classification accuracy (%) on CIFAR-10 and CIFAR-100 datasets with 40% IDN, using multiple combinations of different methods with different pre-trained feature extractors

Method	Pre-trained f_{ext}	CIFAR-10	CIFAR-100
DISC	-	85.61	64.46
CARD	ResNet34	71.77	56.51
LRA-diffusion	ResNet34	86.73	65.55
DLD (Ours)	ResNet34	87.17	66.59
CARD	SimCLR	75.93	59.02
LRA-diffusion	SimCLR	87.13	66.34
DLD (Ours)	SimCLR	89.73	68.57
CARD	ViT-L/14	76.97	61.26
LRA-diffusion	ViT-L/14	93.68	67.67
DLD (Ours)	ViT-L/14	**96.49**	**76.03**

(10%∼20% IDN), the performance of DLD-knn is superior to other combinations. However, as the noise ratio increases, DLD-cos exhibits more stable performance. One possible explanation is that cosine similarity, based on directional similarity, is more resistant to noise interference than traditional Euclidean distance. In medium to high noise environments (30% and above noise level), both DLD-vector, which combines one strategy, and DLD-ws, which combines two strategies, progressively improve model performance. This indicates that the training strategy of vector labels retains more secondary information from the neighborhood label distribution, while the fusion feature method provides the model with more diverse knowledge. Both of these training strategies incorporate the idea of implicit regularization, achieving improved model generalization by enriching learning information. However, in low-noise environments, the interference of redundant information during DLD-vector and DLD-ws training slightly affects model classification performance. Nevertheless, these four training settings show significant performance improvements compared to LRA-diffusion, further demonstrating the necessity and effectiveness of each component of our method. Considering that DLD-ws demonstrates superior performance stability in all experiments, we typically recommend using this training configuration.

G. Detailed Experimental Results on Clothing1M

As shown in Table F.3, on Clothing1M, the performance of DLD combined with ViT-L/14 is slightly lower than that of the combination model of LRA-diffusion and CC. This combination model employs the CC framework for two-stage label filtering on the dataset and extracts a feature space robust to noise, incurring training costs higher than the overhead of the diffusion model itself, consequently yielding a slight improvement in classification performance. Although integrating CC in the same manner into DLD also enhances the model's performance, we argue that this contradicts the original design principle of DLD, which prioritizes efficiency. Such trade-offs are deemed unreasonable for large-scale training tasks.

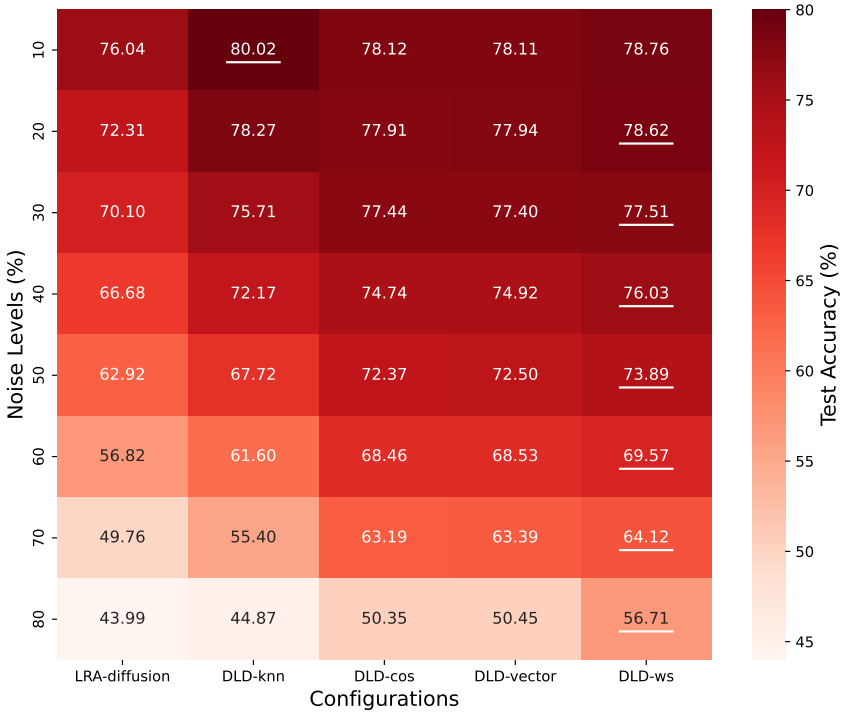


Figure F.1. Test accuracy (%) of LRA-diffusion and different combinations of DLD training strategies on the CIFAR-100 dataset with varying levels of IDN.

Table F.3. Classification accuracies (%) on Clothing1M

CE	Co-teaching	DMI	PLC	LongReMix	C2D	NCR	DivideMix
68.94	69.21	72.46	74.02	74.38	74.58	74.60	74.76

CC	DISC	LRA-diffusion	SANM	DLD	LRA-diffusion+CC	DLD+CC
75.40	74.79	74.46	75.63	75.69	75.70	**75.79**

H. Limitations and future work

In this paper, although we experimentally demonstrate that DLD performs excellently across most real-world datasets. the model has not yet been tested in more complex noise environments, such as imbalanced [7] or OOD noise [8]. In addition, part of the overall performance advantage of DLD can be attributed to the pre-correction method. In future work, we aim to integrate the discriminative and generative paradigms, allowing the generative information from the diffusion model to guide the pre-correction stage in reverse, or using another diffusion model to generate the label information. We also will extend this work to regression tasks or multi-label classification tasks, addressing issues such as numerical label noise and the complexities of multi-label noise. Additionally, this work does not provide a detailed analysis of the independence between diffusion coefficients and diffusion paths. In future research, we will further investigate the impact of the variation patterns of diffusion coefficients on model performance.

References

[1] Jian Chen, Ruiyi Zhang, Tong Yu, Rohan Sharma, Zhiqiang Xu, Tong Sun, and Changyou Chen. Label-retrieval-augmented diffusion models for learning from noisy labels. *Advances in Neural Information Processing Systems*, 36, 2024. 7

[2] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao,

and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11442–11450, 2021. 4

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 9

[4] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018. 7

[5] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022. 7

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3

[7] Yingsong Huang, Bing Bai, Shengwei Zhao, Kun Bai, and Fei Wang. Uncertainty-aware learning against label noise on imbalanced datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6960–6969, 2022. 12

[8] Galadrielle Humblot-Renaux, Sergio Escalera, and Thomas B Moeslund. A noisy elephant in the room: Is your out-of-distribution detector robust to label noise? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22626–22636, 2024. 12

[9] Jongwoo Ko, Sumyeong Ahn, and Se-Young Yun. Efficient utilization of pre-trained model for learning with noisy labels. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023. 7

[10] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 5, 7

[11] Jiawei Liu, Qiang Wang, Huijie Fan, Yinong Wang, Yandong Tang, and Liangqiong Qu. Residual denoising diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2773–2783, 2024. 8

[12] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020. 7

[13] Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016. 4

[14] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[15] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 9

[16] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4

[17] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 6, 8

[18] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. *Advances in neural information processing systems*, 32, 2019. 7

[19] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 4

[20] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 7